Virtual Private Network

Tecnologie e protocolli per Internet II (TPI2)

rev 1.0

Andrea Detti

University of Roma “Tor Vergata”

Electronic Engineering dept.

E-mail: andrea.detti@uniroma2.it

Ringraziamenti: devo un ringraziamento al Prof. Nicola Blefari-Melazzi, al Prof. Stefano Salsano, autori di presentazioni da cui sono alcune delle seguenti slides.

Introduzione

- Allow secure traffic exchange among company branches ditributed over the entire territory
- Usually required by business customers
- Virtual Private Networks
  - » **Private**: allows communication between subnets in different networks as they were in the same private network (as for addressing, routing and security)
  - » **Virtual**: the required links between networks are (necessarily) virutal (not phisical). The support network is not private.

- Indirizzamento IP privato
  - » 10.0.0.0/8
  - » 172.16.0.0/12
  - » 192.168.0.0/16

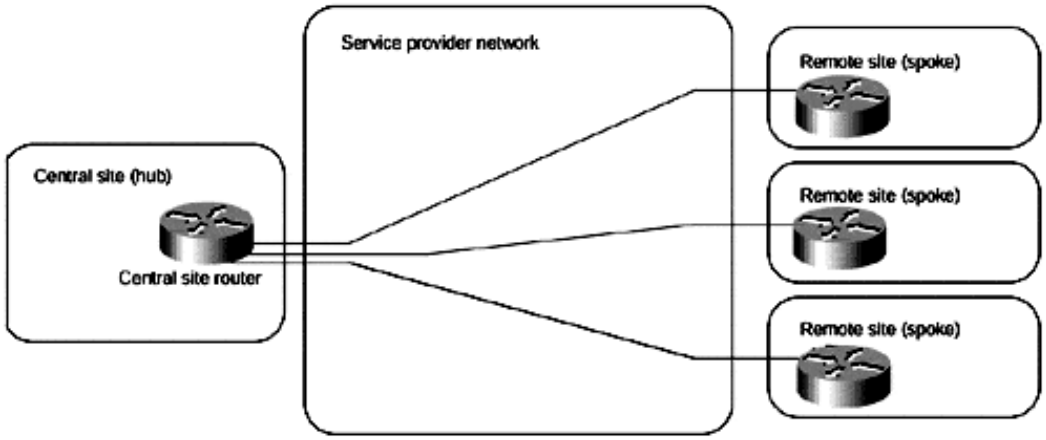
- **Requirements**: unique addressing in a VPN

Modelli di VPN

- **Communication models**
  - » **Intra-company (Intranet)**
  - » **Inter-company (Extranet)**
    - » **Addresses must be unique**
  - » **VPDN (Virtual Private Dialup Network)**
    - » **Dynamica address configuration**
- **Data transfer models**
  - » **Overlay: ISP network is used only for transporting features. Routing information is exchanged between company networks. The VPN topology composed by point to point links configured by the customers**
  - » **Peer-to-peer: the ISP is responsible also for exchangin routing information Logical topology is defined by the customers. Physical topology is defined by the ISP**

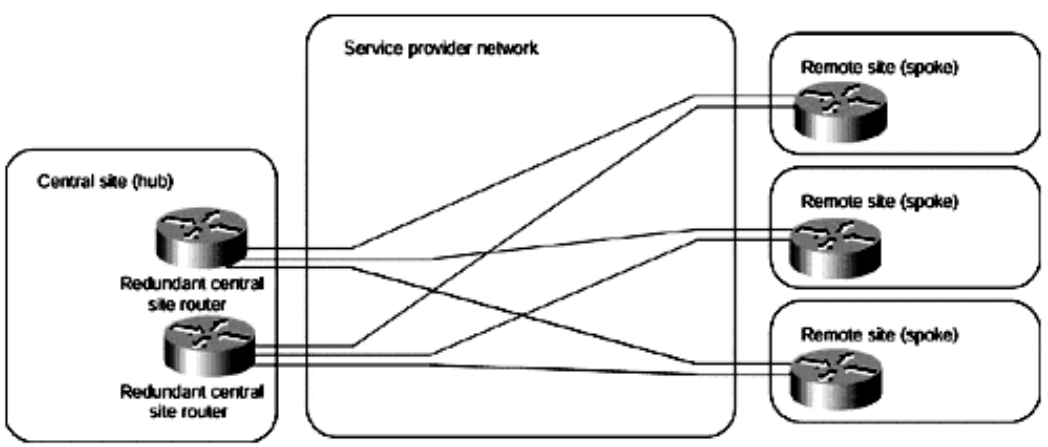
Topologie VPN– Hub and Spoke

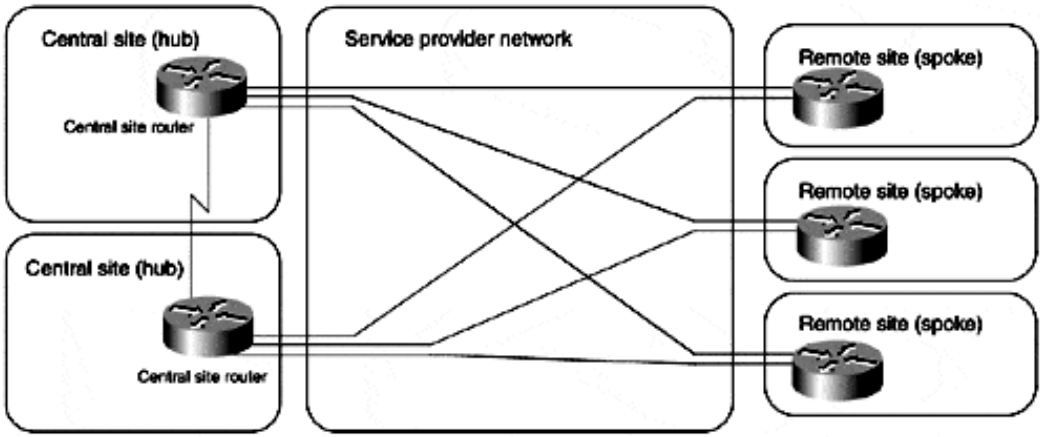
- VPN tpology depends on the specific customers' needs. Nevertheless there are some "standard" topologies…
- Hub-and-spoke Topology:
  - » Remote branches (spoke) connected to a central site (hub).
  - » Spokes can communicate with each other, but inter-spoke should be negligible then spoke-hub traffic



Topologie VPN– Hub and Spoke

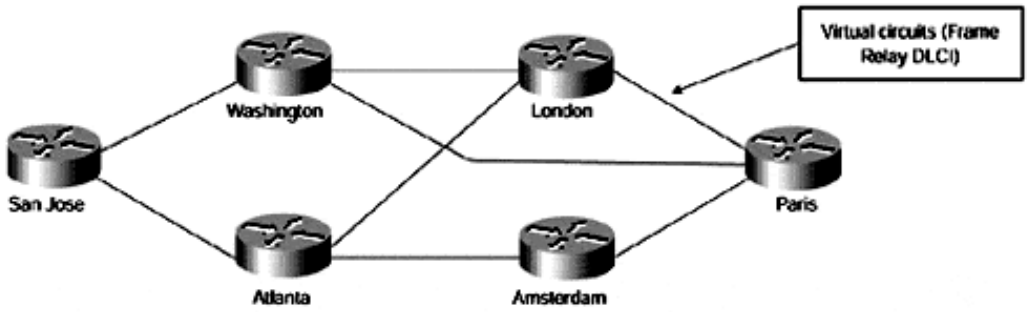
- **Hub Backup**





Topologie VPN– Partial- Full-Mesh

- Quando vi è un cospicuo scambio di dati fra i siti aziendali la topologia Hub-and-Spoke è poco efficace in quanto tutto il traffico spoke-spoke attraversa l'hub che diventa quindi il collo di bottiglia
- In questo caso topologie parzialmente o totalmente connesse sono preferibili
- Business case:
  - » Aziende senza una stretta organizzazione gerarchica
  - » Applicazioni di tipo peer-to-peer (messaging o collaboration system)
  - » Per aziende multinazionali in cui il costo della soluzione hub-and-spoke può essere elevato a causa del costo eccessivo di link internazionali.

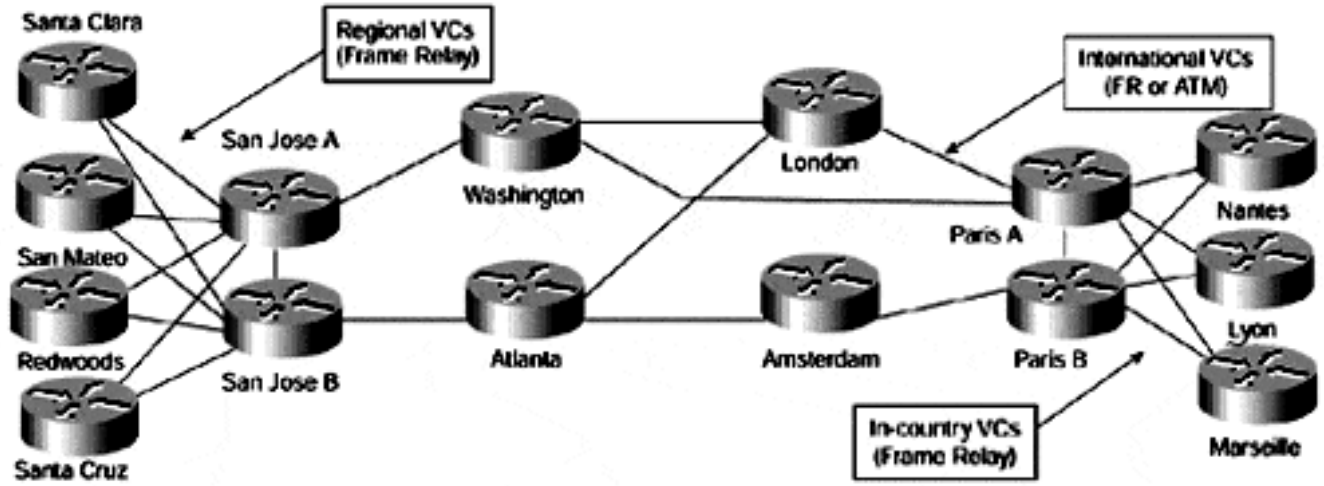


Topologie VPN– Partial- Full-Mesh

- La topologia full-mesh è di facile pianificazione basta avere la matrice di traffico *A(i,j)=x* Mbps e chiedere all'ISP un collegamento fra il sito *i* ed il sito *j con x* Mbps
- Il costo full-mesh può essere elevato poiché il numero di collegamenti affittati è n*(n-1)
- Pertanto si opta spesso per una partial-mesh
- Approccio di pianificazione topologica di una partial mesh
  - 1) Creare una topologia connessa attraverso collegamenti solo fra sedi che hanno un elevato scambio di dati
  - 2) Dalla matrice di traffico ed assumendo un routing shortest-path calcolare l'ammontare di banda richiesta su tutti i collegamenti installati
  - 3) ordinare i collegamenti all'ISP + economico ;-)

Topologie VPN– Hybrid

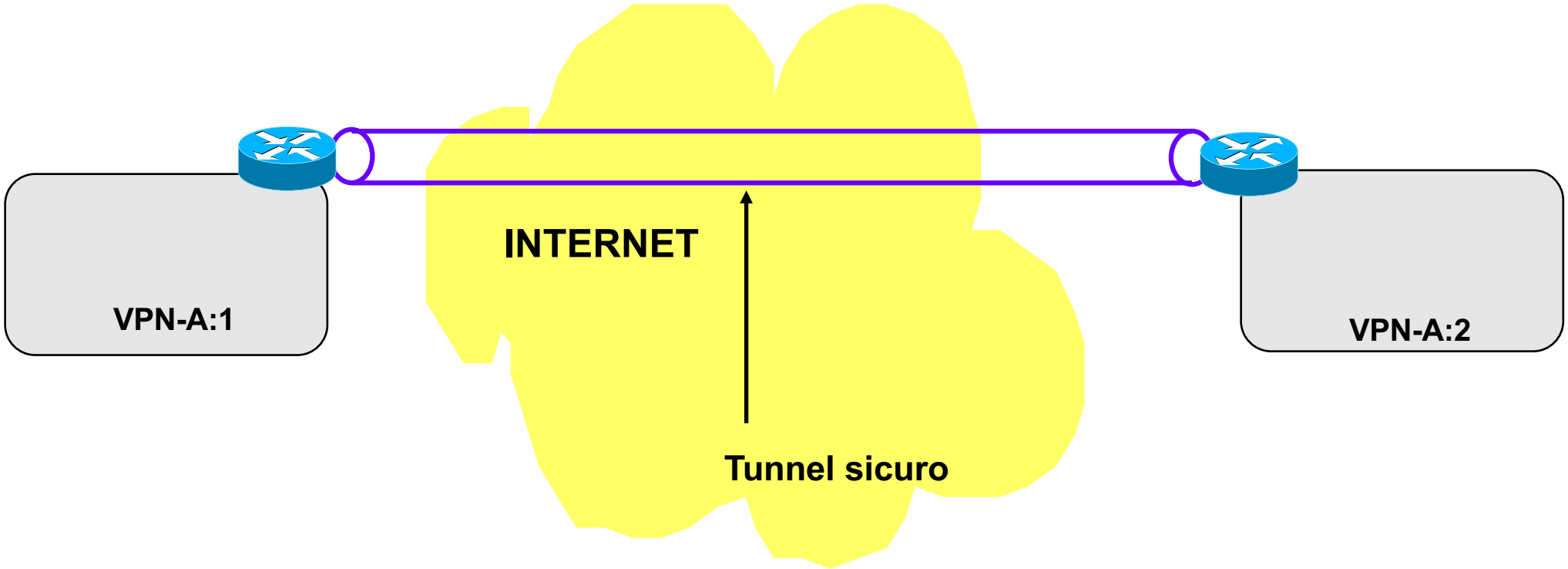
- VPN molto grandi internazionali sono spesso composte da VPN nazionali di tipo hub-and-spoke e la parte internazioanle (backbone) è una partial-mesh fra gli hubs



Overlay VPN

Concetti di base

- Sono VPN realizzate su rete pubblica INTERNET.
- La sicurezza delle comunicazioni deve essere realizzata ent-to-end, in quanto non ci si può fidare di una rete sicura di trasporto (e.s., MPLS)
- Il modello di riferimento prevede la creazione di Tunnel sicuri fra gli end-points della VPN



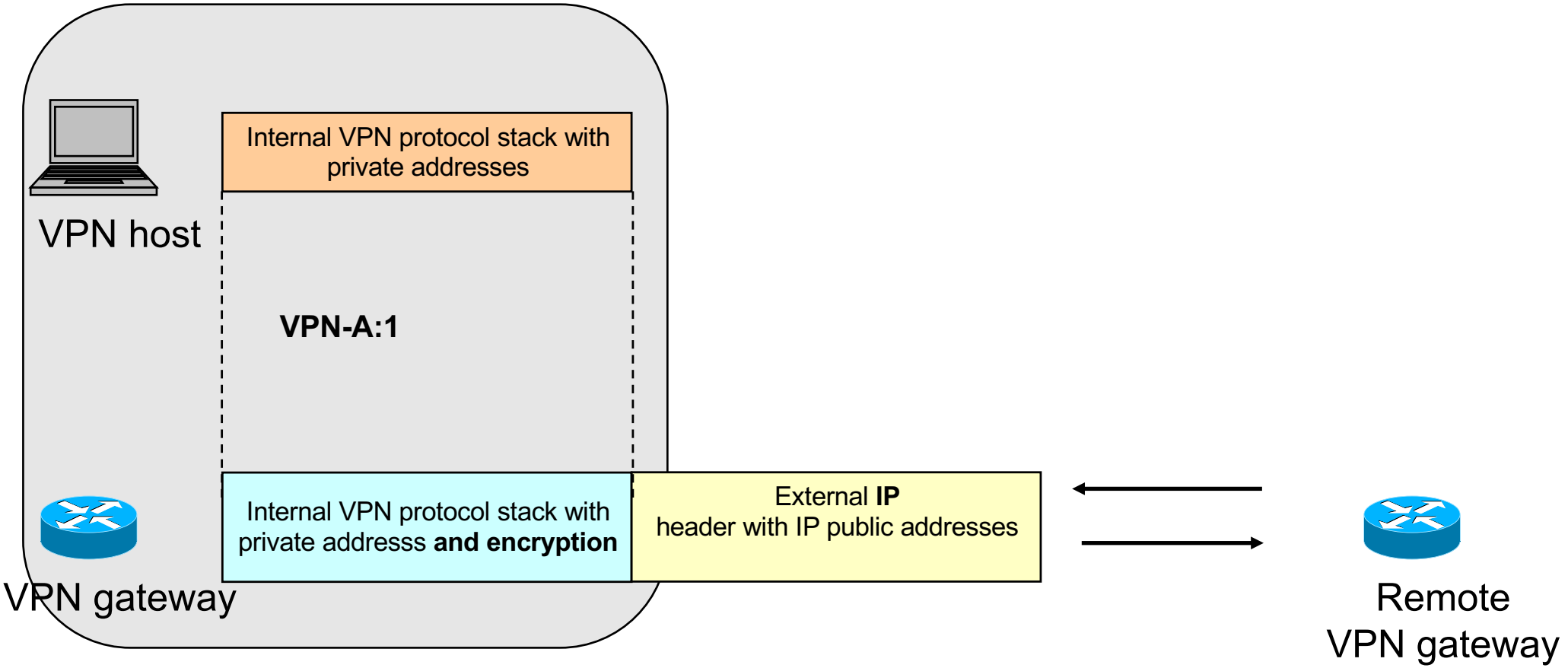
Tunneling

- Il termine tunneling si riferisce a un insieme di tecniche per cui un protocollo viene incapsulato in un protocollo dello stesso livello o di livello superiore per realizzare configurazioni particolari.

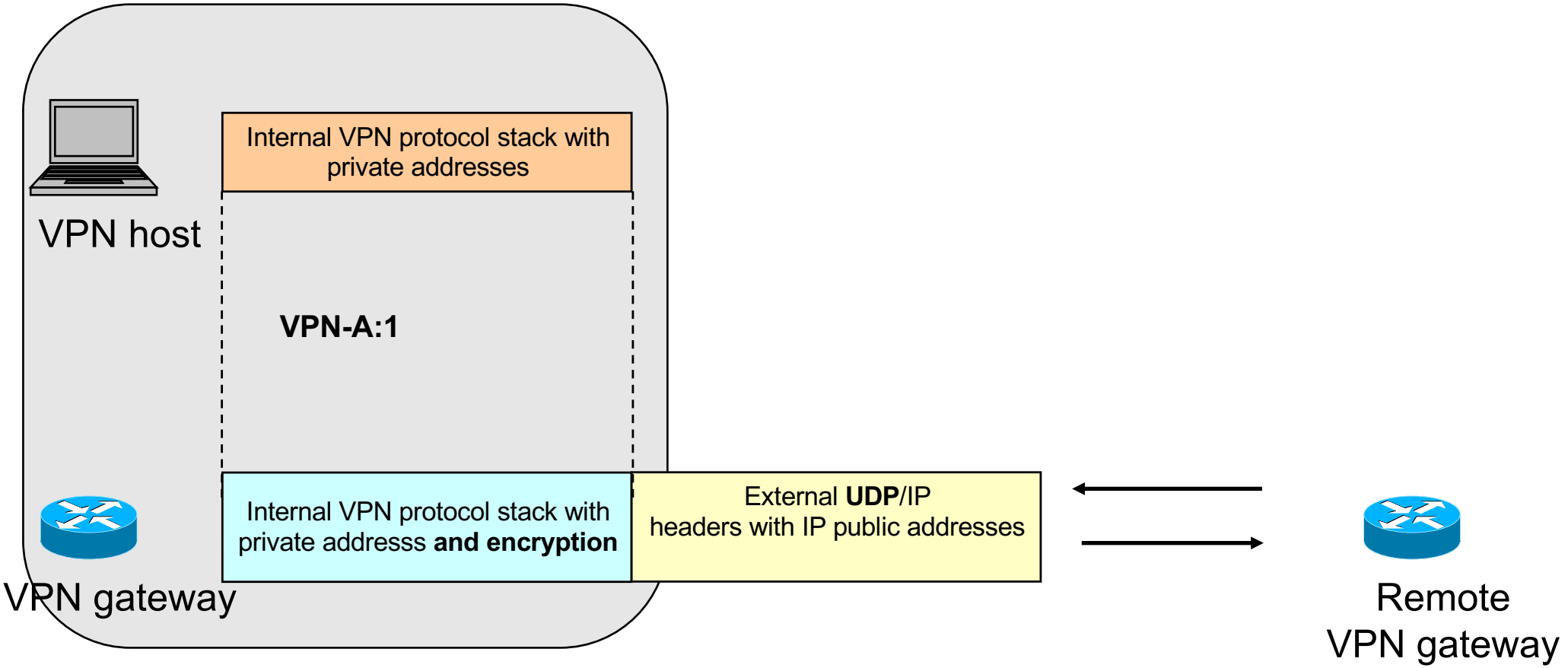
- Le due tipologie di tunnel più importanti sono
  - Tunnel IP
  - Tunnel UDP

- La sicurezza della comunicazione è ottenuta attraverso la cifratura dei dati che sono inseriti all'interno del Tunnel e la autenticazione degli end-point.
  - Sicurezza delle Reti (Prof. Giuseppe Bianchi)

IP tunnel


Internal VPN protocol stack with private addresses
VPN host
VPN-A:1
Internal VPN protocol stack with private addresss **and encryption**
External **IP** header with IP public addresses
VPN gateway
Remote VPN gateway

UDP tunnel

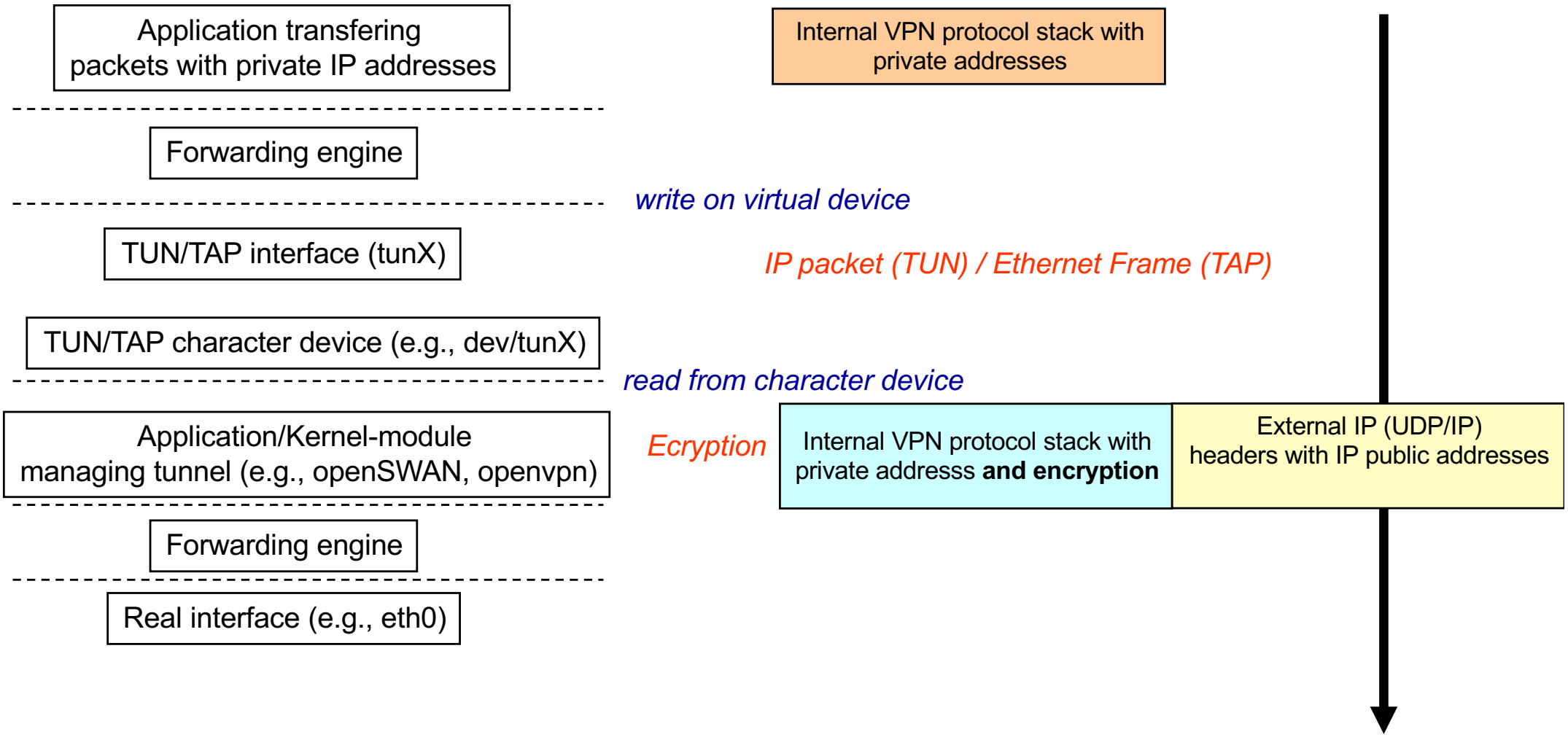

Internal VPN protocol stack with private addresses
VPN host
VPN-A:1
Internal VPN protocol stack with private addresss **and encryption**
External **UDP**/IP headers with IP public addresses
VPN gateway
Remote VPN gateway

Tunnel visibili dal livello applicativo: TUN / TAP driver

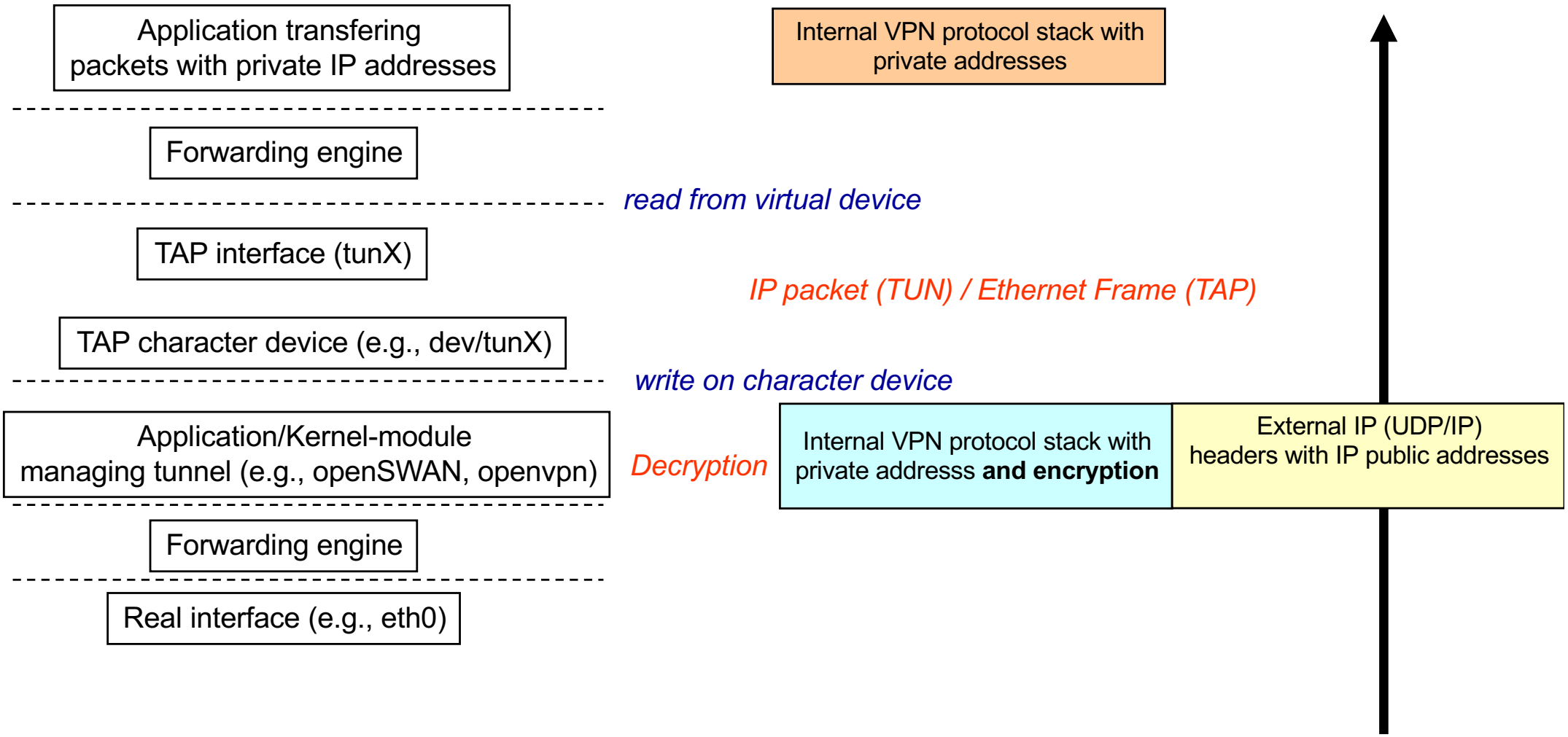
- **Un tunnel può essere reso visibile o meno al livello applicativo**
- **Quando visibile, è spesso presentato come una scheda *virtuale* layer-3 che trasferisce/riceve pacchetti IP (TUN driver) o come una scheda *virtuale* layer-2 (TAP driver) che trasferisce/riceve trame Ethernet**
- **Con un TUN si trasferiscono pacchetti (Routed VPN)**
- **Con un TAP si trasferiscono trame Ethernet (Bridged VPN)**

- **Oltre alla scheda virtuale un TUN/TAP driver può ricevere/trasmettere pacchetti su una zona comune di memoria (*character device*)**
- **Una applicazione che scrive sul *character device* invoca una *ricezione* sul *device di rete virtuale***
- **Una applicazione che scrive sul *device di rete virtuale* invoca una *ricezione* sul *character device***

Trasferimento di un pacchetto via TUN/TAP driver



Ricezione di un pacchetto via TUN/TAP driver



User-space VPN

Overlay VPN

User-space VPN

- **Sono VPN i cui link sono dei tunnel UDP o TCP gestiti da uno specifico tool di livello applicativo**
- **La sicurezza su questi tunnel è garantita da (Datagram) Transport Layer Security TLS (DTLS)**
- **Si chiamano user-space VPN poichè sono basate sui socket che sono controllabili dallo user-space**
- **UDP vs TCP tunnel: UDP**
  - **Il tunnel deve transportare lo stack protocollare TCP-UDP/IP**
  - **questo stack è stato ottimizzato per un trasporto diretto su una rete non *reliable*, quale quella Internet. Pertanto, fare un tunnel UDP (i.e., unreliable) è preferibile in quanto il tunnel ha le stesse proprietà di un trasporto diretto su Internet, a parte un overhead addizionale.**

User-space VPN – Packet handling

- I pacchetti IP (Ethernet) trasferiti dall'applicazione su una scheda virtuale TUN/TAP sono trattati dal tool che gestisce la "*user-space VPN*" (e.g., openvpn)
- Alla ricezione di un pacchetto proveniente da una applicazione locale, il tool *user-space VPN* controlla l'indirizzo IP di destinazione e decide su quale socket UDP (TCP) incapsulare il pachetto cifrato
- Pertanto il tool *user-space VPN* possiede una sua tabella di routing (overlay) svincolata dalla tabella di routing dell'OS

- Le entry di questa tabella *overlay* sono del tipo
  <netid, mask, public_ip_da, udp_port>

- Il tool *user-space VPN* remoto decifra, autentica e decapsula i pacchetti IP (Ethernet) entranti e li inietta (in su) sul TUN/TAP driver (i.e., scrive sul TUN/TAP character device)
- Il forwarding dello OS provvederà alle successive operazioni di forwarding intra-VPN

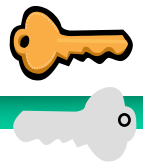
Routed o Bridged VPN ?

- **Quale stack protocollare è trasportato dal tunnel ?**
- ***Bridged VPN:* tap driver che trasporta trame ethernet**
  - » **Per gli host connessi alla VPN, la VPN è un dominio Ethernet, pertanto si può parlare di una *Virtual wide-area LAN***
  - » **Il traffico Broadcasts si trasmette sulla VPN – questo permette il funzionamento di software che dipendono da una LAN sottostante (e.s., Windows NetBIOS file sharing and network neighborhood browsing).**
  - » **Nessun routing da configurare**
  - » **Problema: su reti molto grandi, il trasferimento del Broadcast pone dei seri limiti alla scalabilità.**
- ***Routed VPN*: tun driver che trasporta pacchetti IP**
  - » **Ogni collegamento è una subnet ip diversa → no broadcast traversal**

Cenni di TLS

User-space VPN

Public Key Cryptography



Bob

Bob's Public Key

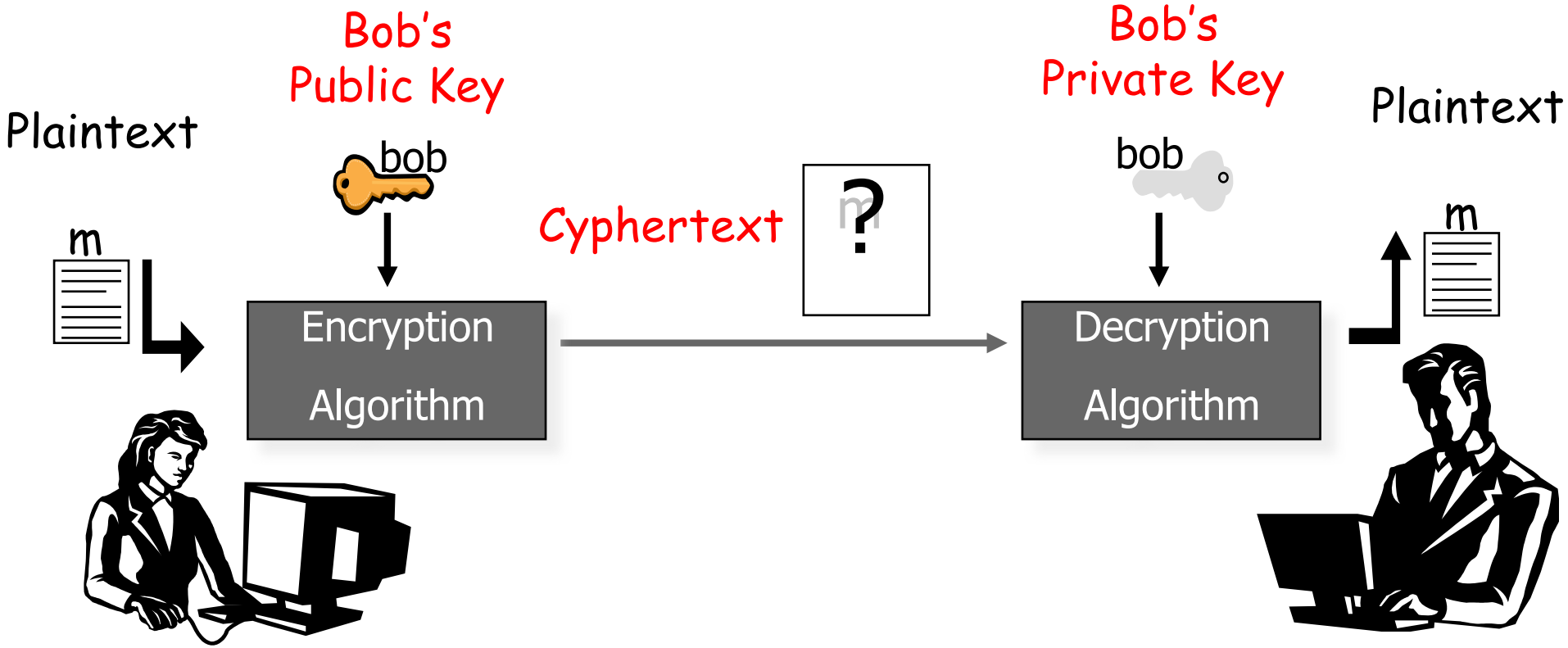
Bob's Private Key

Alice

Bob's Public Key

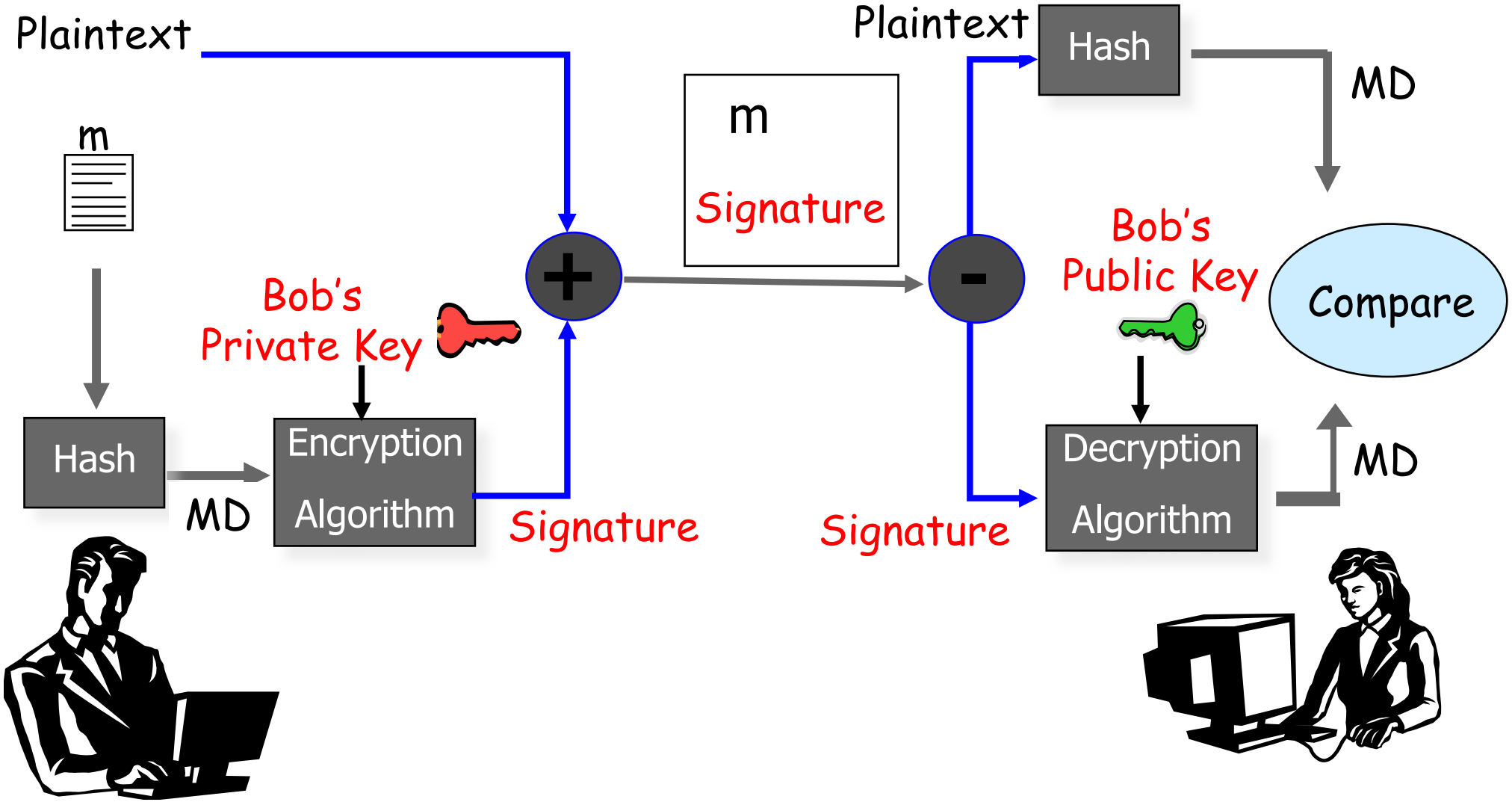
Public Key Cryptography

Confidentiality

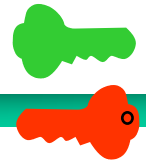


Public Key Cryptography

Authentication & Message Integrity Check



PK Cryptography: Digital Certificate



Come può Alice essere sicura che la Bob's public key è autentica?

Certificate Authority Center

Bob Info:

- Name
- Department
- Cubical Number

Certificate Info:

- Expiration Date
- Serial Number

Bob's Public Key

Sign data

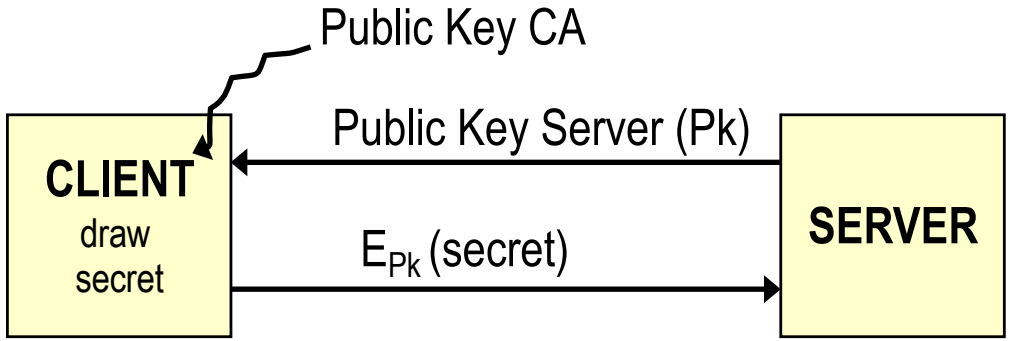
CA's Private Key

Transport Layer Security

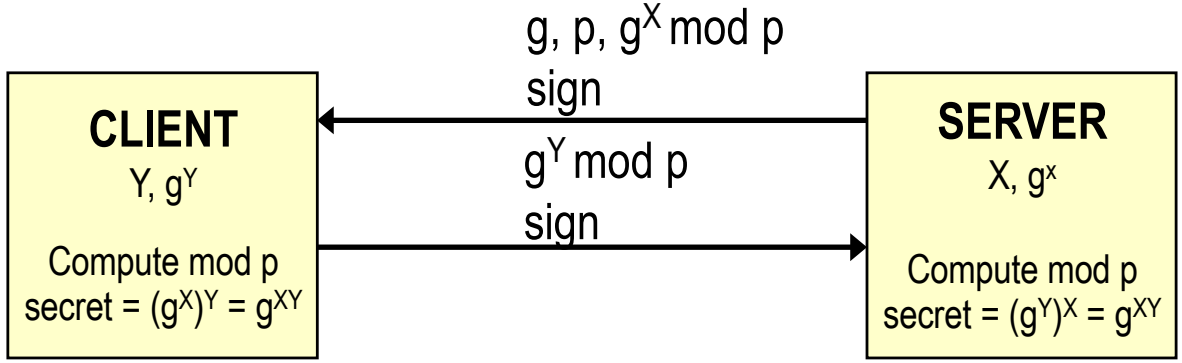
- Gestisce la comunicazione sicura tra un client ed un server
- Problema: l'uso di una chiave asimmetrica durante uno scambio informativo ad elevato bit/rate può creare un collo di bottiglia sul processing
- Approccio risolutivo utilizzo dei meccanismi a chiave asimmetrica per configurare una chiave simmetrica su entrambi i lati
  - » Key transport (e.g. RSA)
  - » Key agreement (e.g., Diffie-Hellman)

The two basic approaches to key management

- **Key transport (e.g., RSA)**
  - » **Il client genera una chiave random (Secret)**
  - » **La chiave è trasferita al server cifrandola con la chiave pubblica del server**



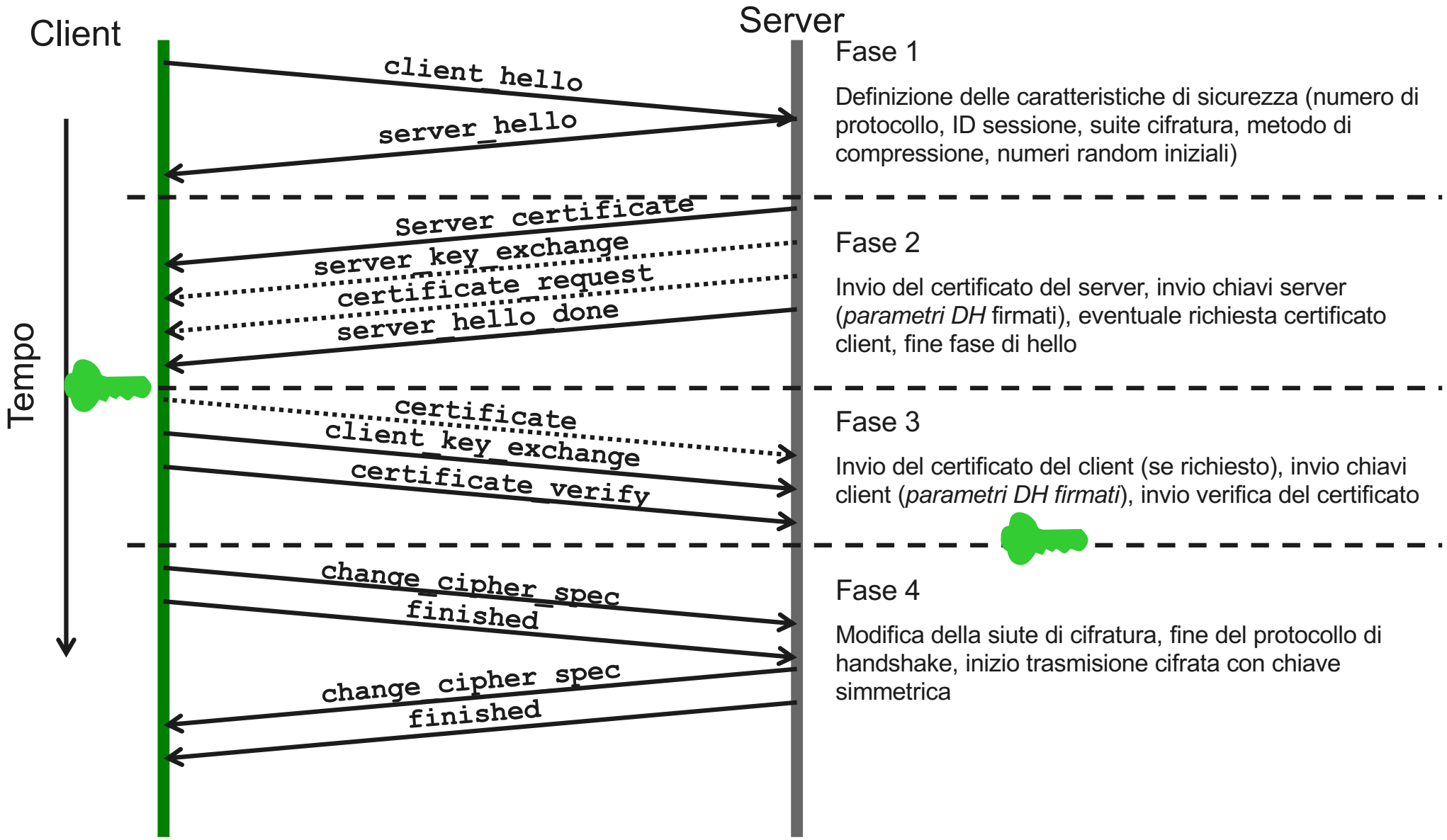
- **Key agreement (e.g., DH Ephermal)**
  - » **Segreto condiviso calcolato da entrambe le parti attraverso lo scambio opportuno di parametri crittografici**



Transport Layer Security (E-DH)

- **TLS / Ephermal Diffie-Hellman :**
  - **Si instaura una connessione sicura tra le parti**
  - **Nella fase di instaurazione, le parti si scambiano i loro certificati, firmati da una CA affidabile in modo che possano autenticarsi a vicenda**
  - **Inoltre, sempre nella fase di instaurazione, le parti si scambiano alcuni parametri (*firmati*) che permettono di decidere quale sia la chiave di cifratura simmetrica da utilizzare durante il successivo trasferimento dati (Ephermal Diffie-Hellman parameters)**
  - **L'intercettazione dei Diffie-Hellman parameters che transitano in rete non permette ad un ascoltatore intruso di capire quale sarà la chiave simmetrica che sarà adottata**
  - **La *firma* su questi Diffie-Hellman parameters assicura le parti che chi sta trasmettendo è effettivamente chi si dichiara di essere**

TLS Handshake (E-DH)



TLS cosa serve e dove (E-DH)

Client1

Certificato client1 firmato dalla CA: è la carta d'identità da trasferire per farsi riconoscere; contiene la chiave pubblica di *client1*

Certificato della CA: contiene la chiave pubblica della CA *con la quale posso validare i certificati che ricevo*

Chiave privata client1 (secret)

Server

Certificato server firmato dalla CA: è la carta d'identità da trasferire per farsi riconoscere; contiene la chiave pubblica del server

Certificato della CA: contiene la chiave pubblica della CA *con la quale posso validare i certificati che ricevo*

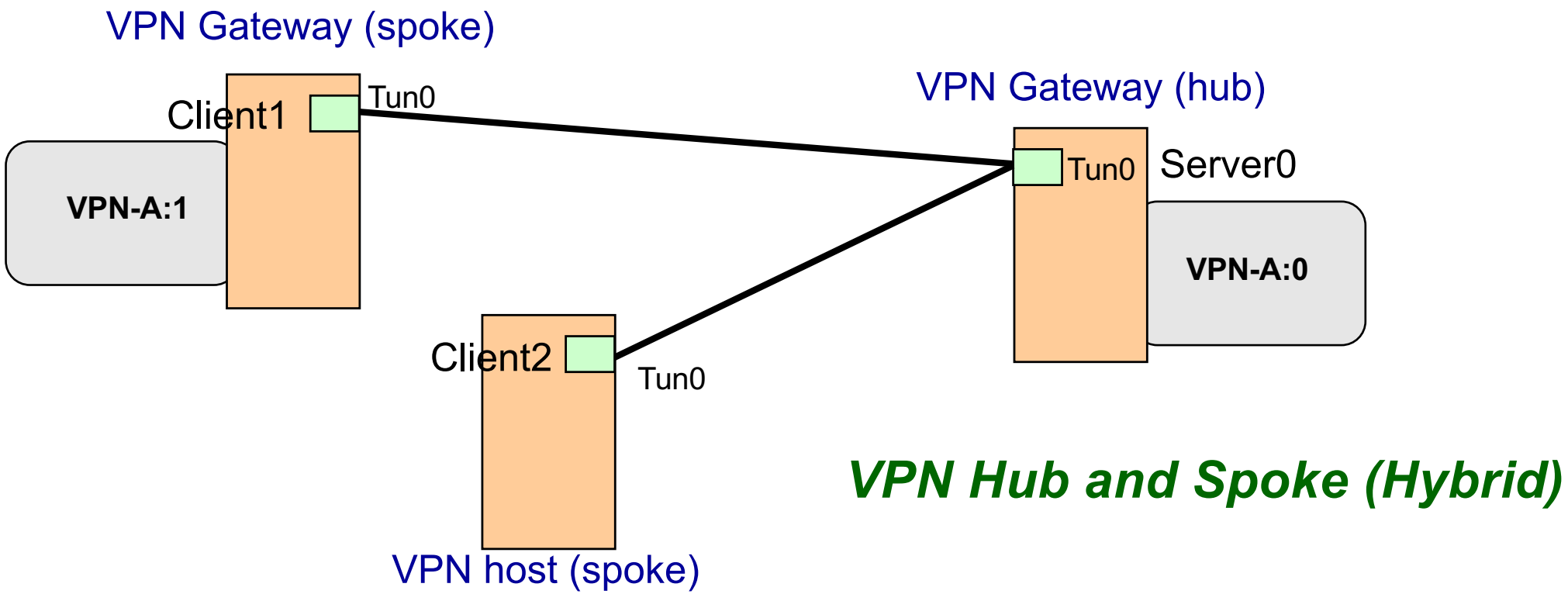
Chiave privata server (secret)

Parametri iniziali Diffie-Hellman

Nel caso in cui non si intende autenticare in client via TLS, il client ha solo bisogno del certificato della CA (*oppure accetta di non verificare la validità dei certificati del server*)

Topologie delle User-Space VPN

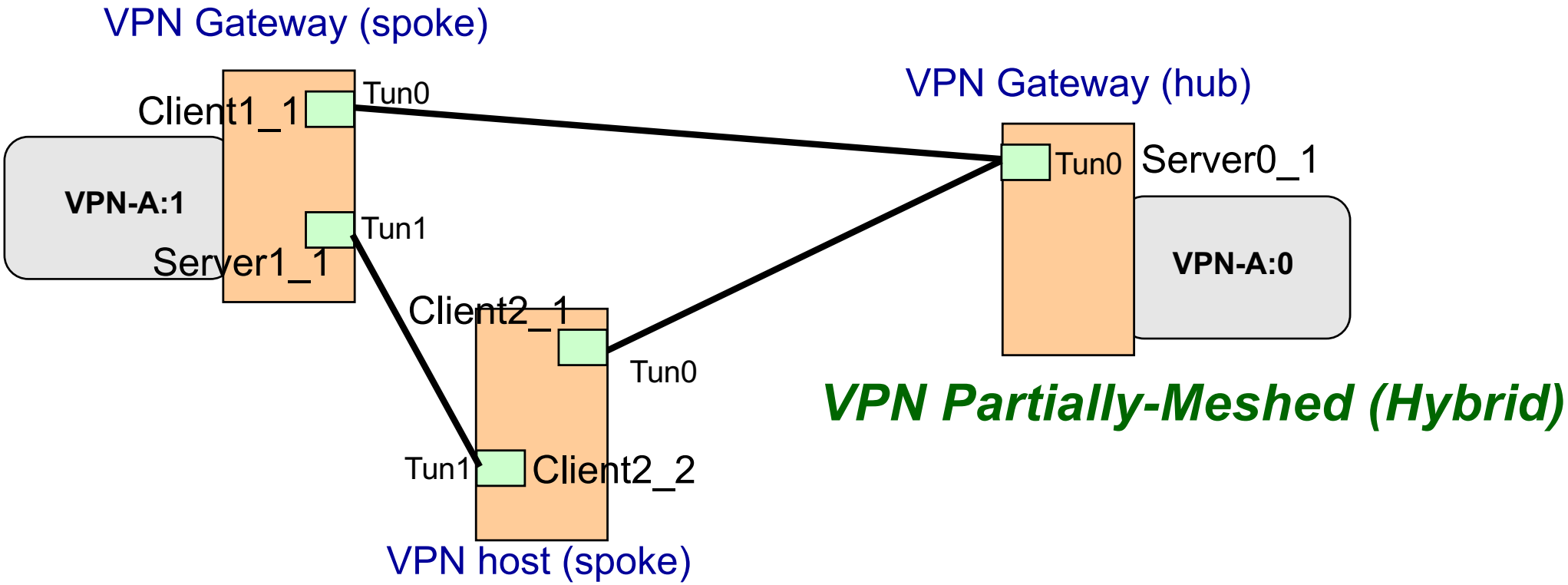
- **Essendo basate su socket, sono conformi al paradigma client-server.**
- **Pertanto, la topologia nativa di questo tipo di VPN è Hub-and-Spoke dove l'Hub è il server e gli spokes sono i client**
- **Il client ed il server possono girare sia host, che gateway**
  - » **Host-to-Host VPN: VPN overlay in cui i tunnel terminano su host**
  - » **Gateway-to-Gateway VPN overlay i cui tunnel terminano su gateway di reti private (unica tipologia offerta da MPLS MB-iBGP)**
  - » **Hybrid: soluzione ibrida; e.s. host mobile che si connette alla LAN aziendale via VPN gaeway**



VPN Hub and Spoke (Hybrid)

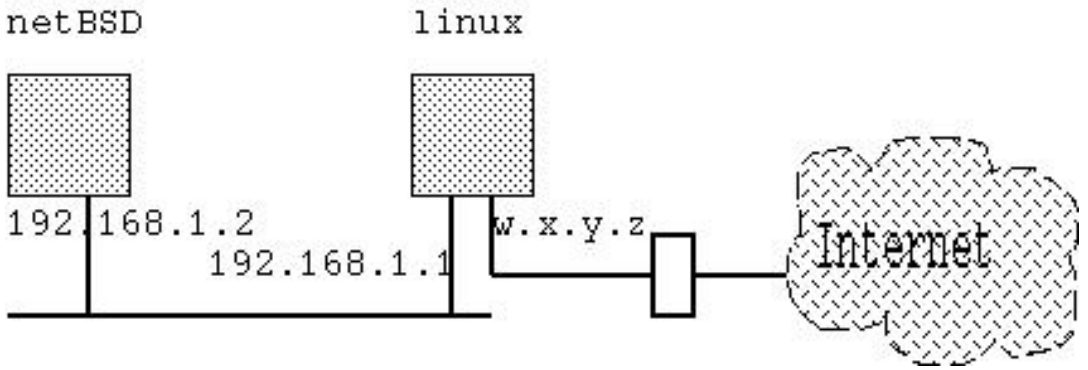
Topologie delle User-Space VPN

- **Topologie meshed richiedono combinazioni client-server e/o più clients sulla stessa macchina**
- **Inoltre, richiedono una buona cura della configurazione delle tabelle di instardamento IP / overlay**



User-space VPN vs NAT e Dynamic IP

- Molto spesso un piccolo ufficio o una abitazione sono connessi ad Internet attraverso un gateway ADSL che ottiene un indirizzo pubblico IP dinamico, assegnato dall'ISP alla connessione fisica.
- Per far accedere ad Internet gli hosts interni dietro al gateway, sul gateway è abilitata la funzionalità di Network (and port) Address Translation (NAT o NAPT)



- Basic NAT
  - 192.168.1.2 → w.x.y.z
- NAPT
  - 192.168.1.2, Source Port A →w.x.y.z, Source Port B

User-space VPN vs NAT e Dynamic IP

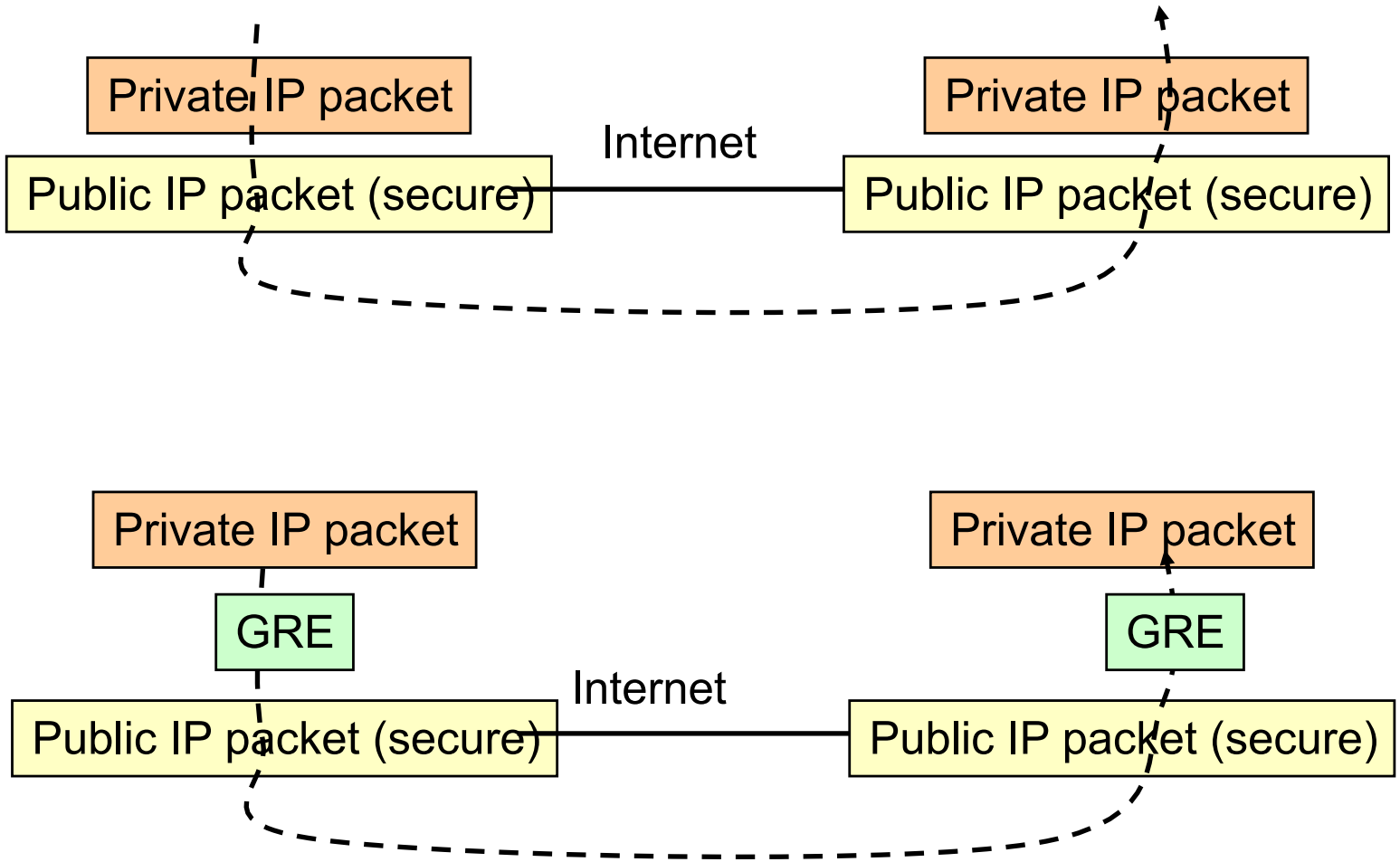
- **Client VPN dietro NAT: nessun problema poiché i tunnel sono basati su socket che interlavorano perfettamente con il NAT del gateway**
- **Server VPN dietro NAT:**
  - **I client VPN necessitano di raggiungere il server. Pertanto i server deve essere raggiungibile attraverso un indirizzo IP pubblico noto ai client**
  - **Per far ciò, il gateway ADSL deve avere anche un indirizzo mnemonico (e.s., srdserver.it) ottenibile da un gestore di *dynamic DNS address* (e.s., www.dyndns.com) e registra sul server DNS l'indirizzo IP pubblico associato all'indirizzo mnemonico ogni volta che l'IP address cambia**
  - **Il gateway del server deve essere configurato in modo da effettuare il port forwarding della porta TCP/UDP del server VPN sull'host che ospita il server VPN**
  - **Il client VPN, nella sua configurazione, ha come indirizzo del server l'indirizzo mnemonico piuttosto che quello IP.**

IPsec VPN

Overlay VPN

IPsec VPN

- **2 modes: i) IP over IP oppure ii) IP over GRE over IP**



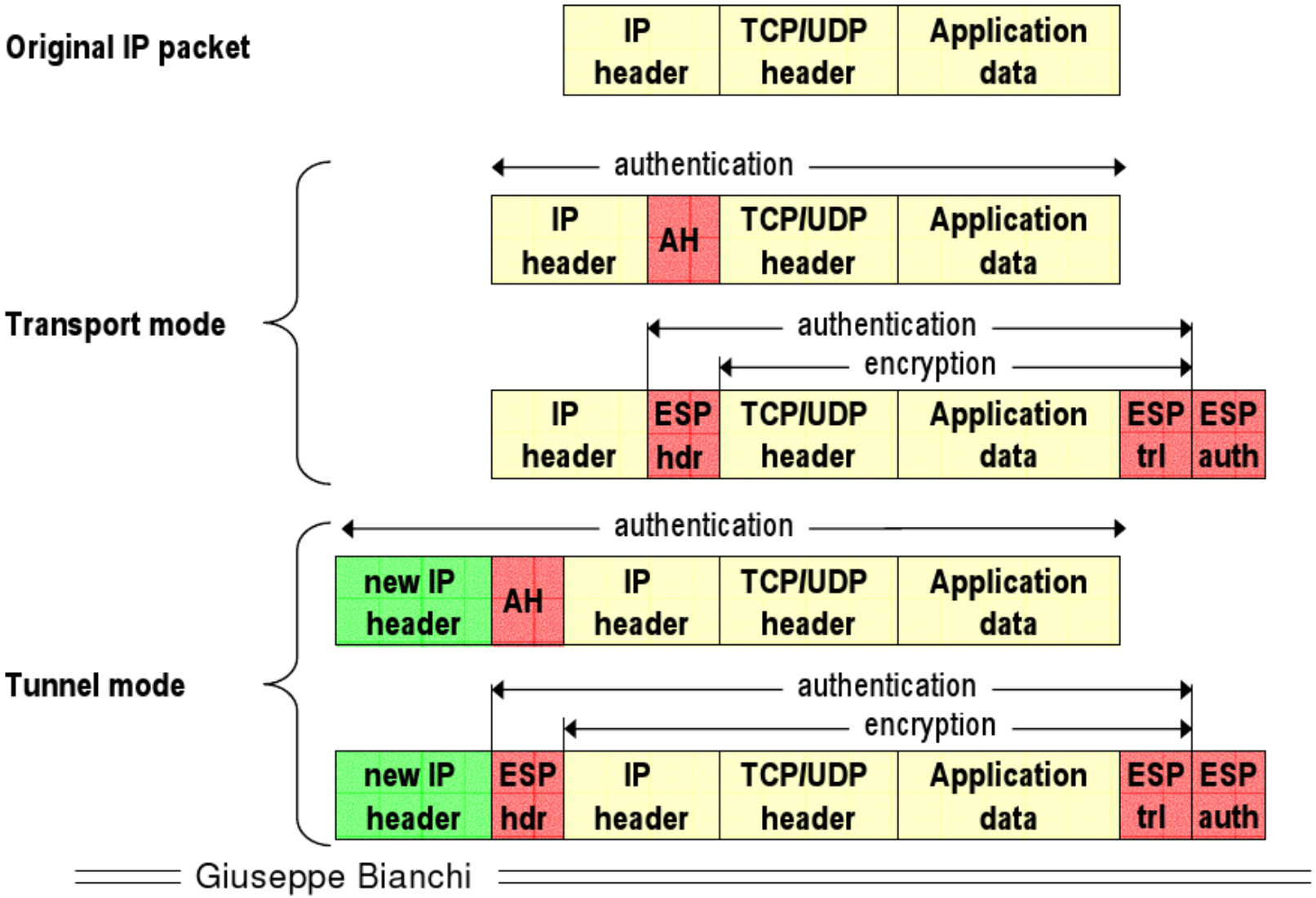
IPsec VPN

- IPsec: protocol suite implementing crittography mechanisms at network layer
  - » Encryption, MAC, authentication
- Su Linux (Windows) in implememted in the OS Kernel
  - » No tun/tap virtual interface
  - » IP sec module is placed between the IP layer and the NIC driver
    - » (without IPSec) IP→Ethernet;
    - » (with IPSec) IP→IPSec→Ethernet
  - » VPN not visible from user space
- 2 encapsulation mode: Transport mode , Tunnel mode

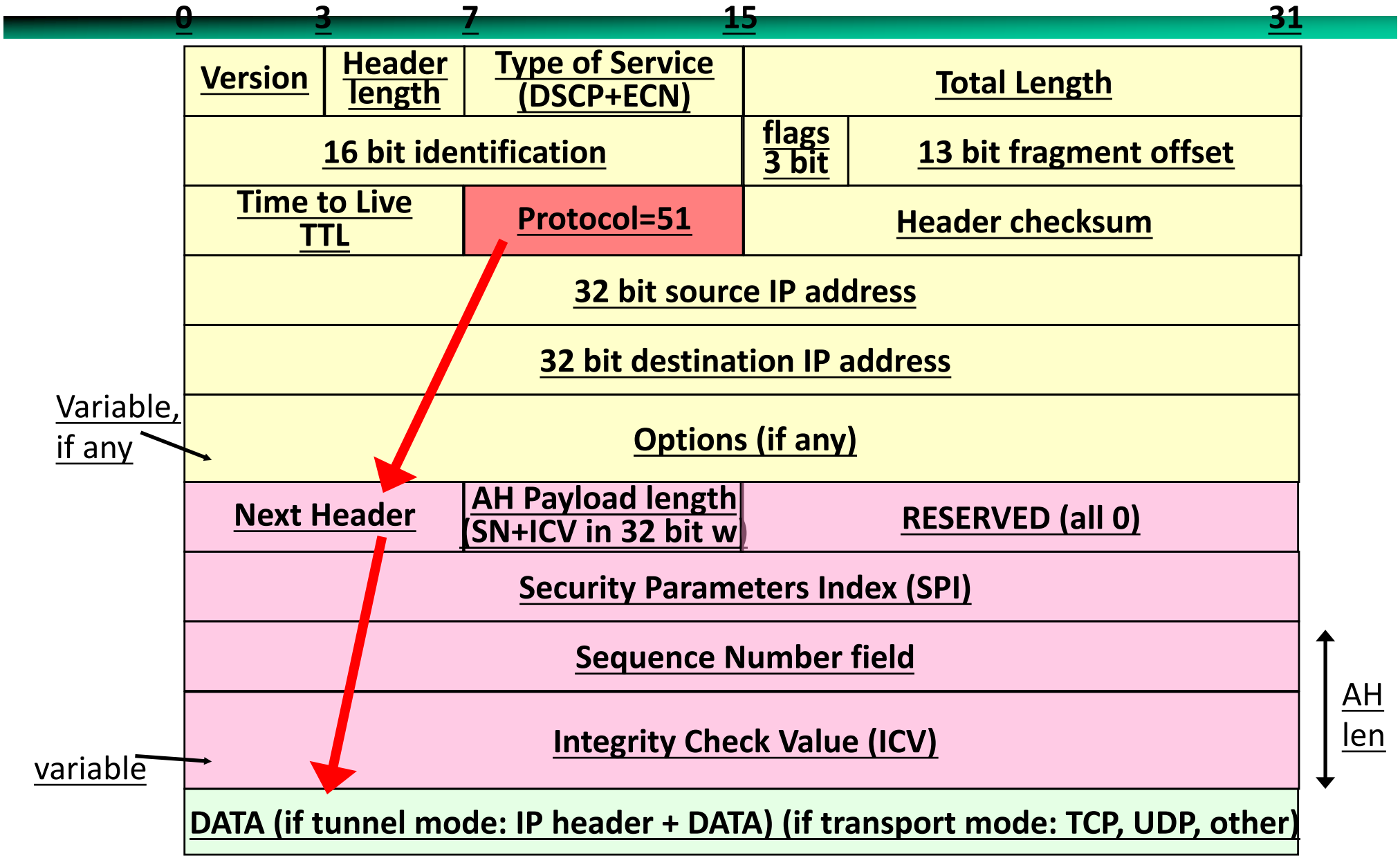
AH and ESP

- Authentication Header
  - » Integrity and data origin authentication
    - » Authentication covers both payload and parts of IP header that do not modify in transfer
  - » Protection against replays
    - » Optional, through extended sequence numbers
- Encapsulated Security Payload
  - » Same services as AH
    - » Though authentication limited to IP payload
  - » Confidentiality through encryption
  - » Traffic flow confidentiality
    - » Improved privacy against eavesdropping
    - » Through padding and dummy traffic generation

Transport vs Tunnel – AH and ESP

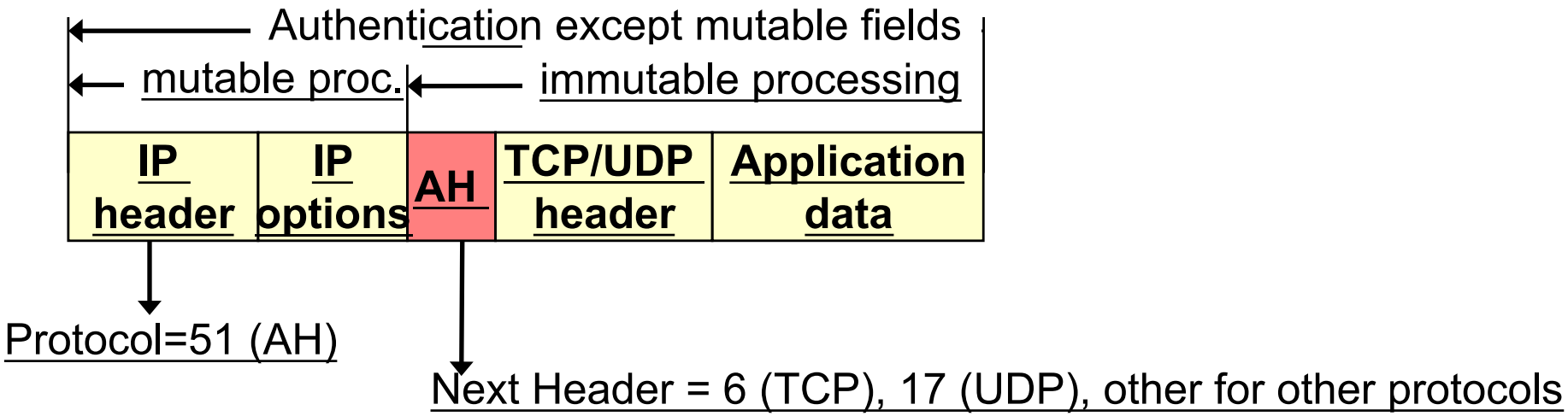


Authentication Header

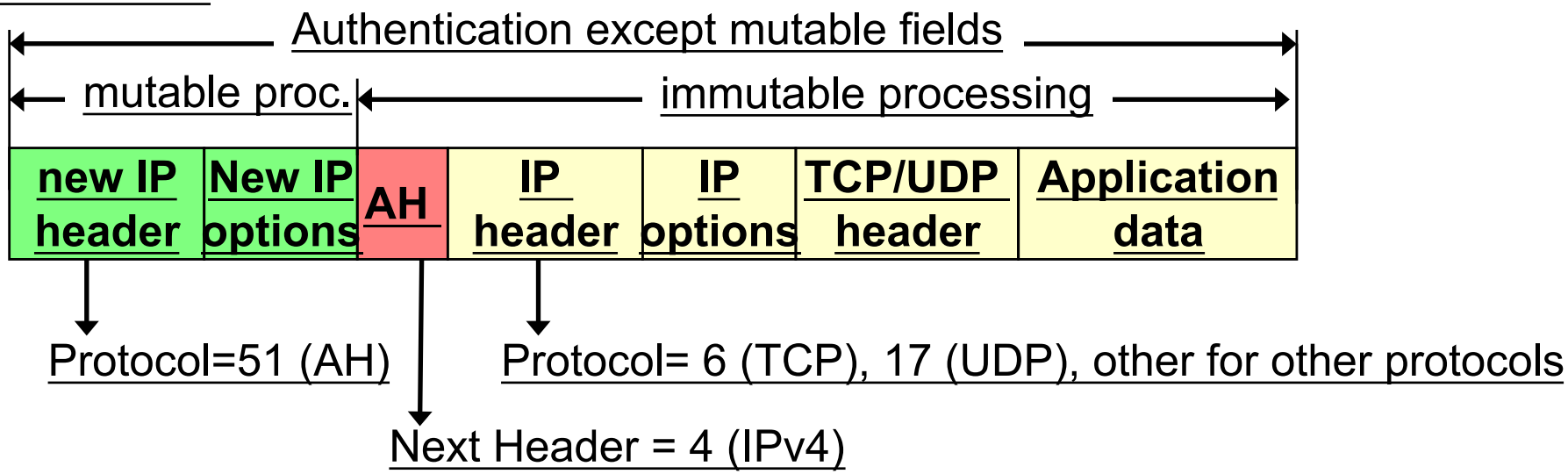


Transport mode, tunnel mode

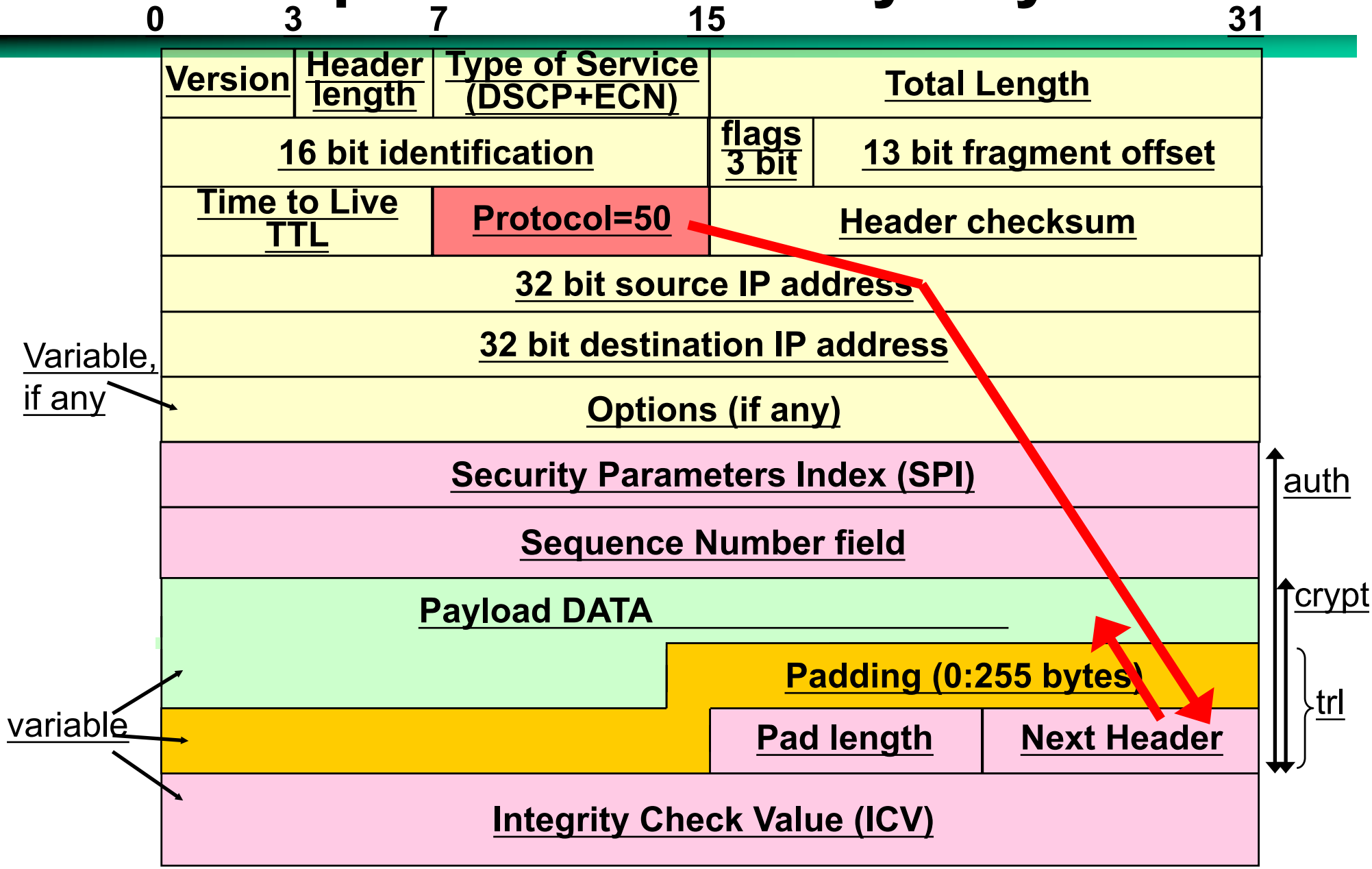
Transport mode:



Tunnel mode:

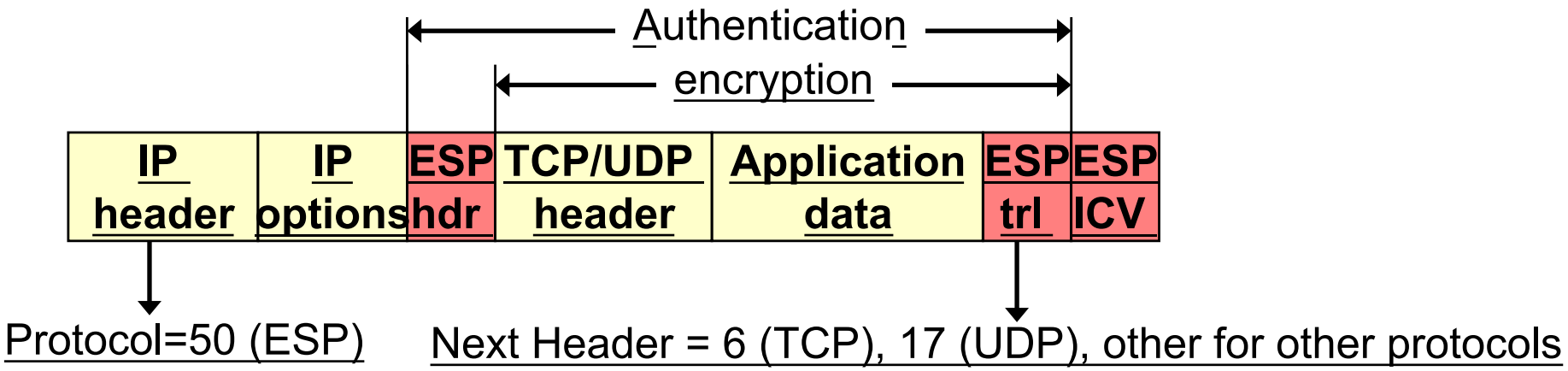


Encapsulated Security Payload

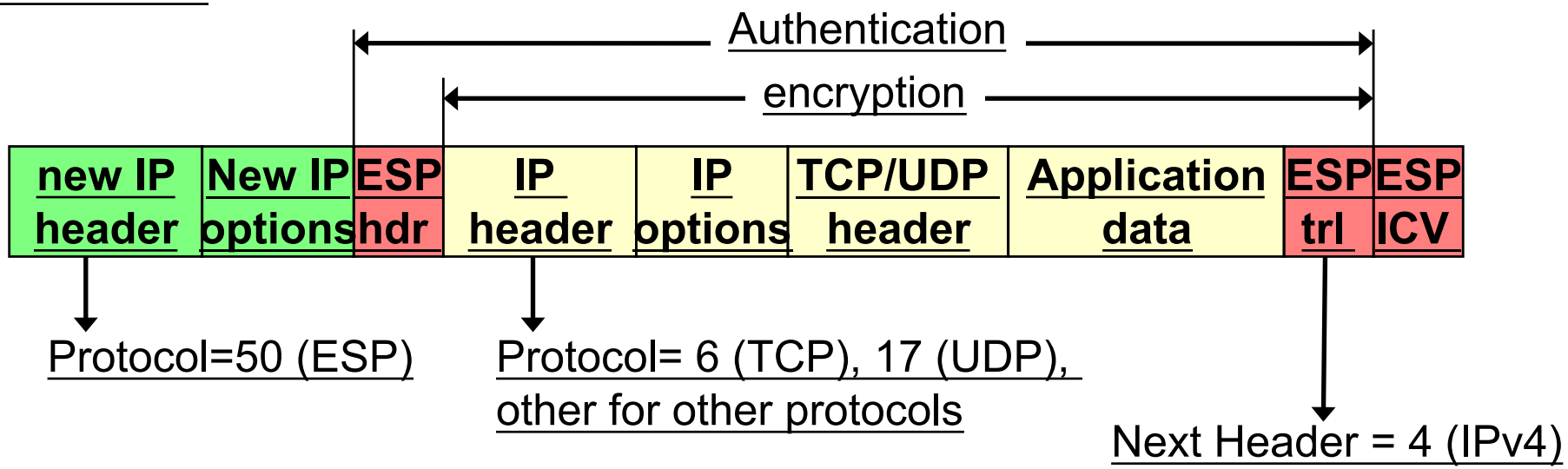


Transport mode, tunnel mode

Transport mode:



Tunnel mode:



IPSEC basics

- SA: Security Association
- Monodirectional logical association bewtween 2 hosts which describes how IPsec protects the communication between 2 devices
  - SPI: Security Parameter Index. Numero che identifica la SA in modo univoco su entrambe gli end-devices
  - AH parameters
  - ESP parameters
  - SRC IP address (local device)
  - DST IP address (remote device)
  - Ipsec mode (transport/tunnel)
- SAD: Security Association Database

Elementi fondamentali di IPsec

- **SP: Security Policy.**
- **Rule that specify if a packet has to be processed by Ipsec**
- **A SP contains:**
  - **Net_id/mask sorgente (in transport mode is the src IP address)**
  - **Net_id/mask destinazione (in transport mode is the dstIP address)**
  - **Src/dst port**
  - **Direction(in out)**
  - **Action (ipsec/discard/none)**
  - **Security protocol (ah/esp/ipcomp)**
  - **Ipsec mode (transport/tunnel)**
  - **Tunnel SRC/DST (tunnel mode)**

- **Ipsec does not consider miulticast broadcast addresses**

- **SPD: Security Policy Database**

Processing IPSec

OUTPUT PACKET

1. **Check SPD for a matching Policy**
2. **If an “Ipsec” policy is found the packet is passed to the SAD**
3. **If a static SA match is found the packet is processed according the matched SA**

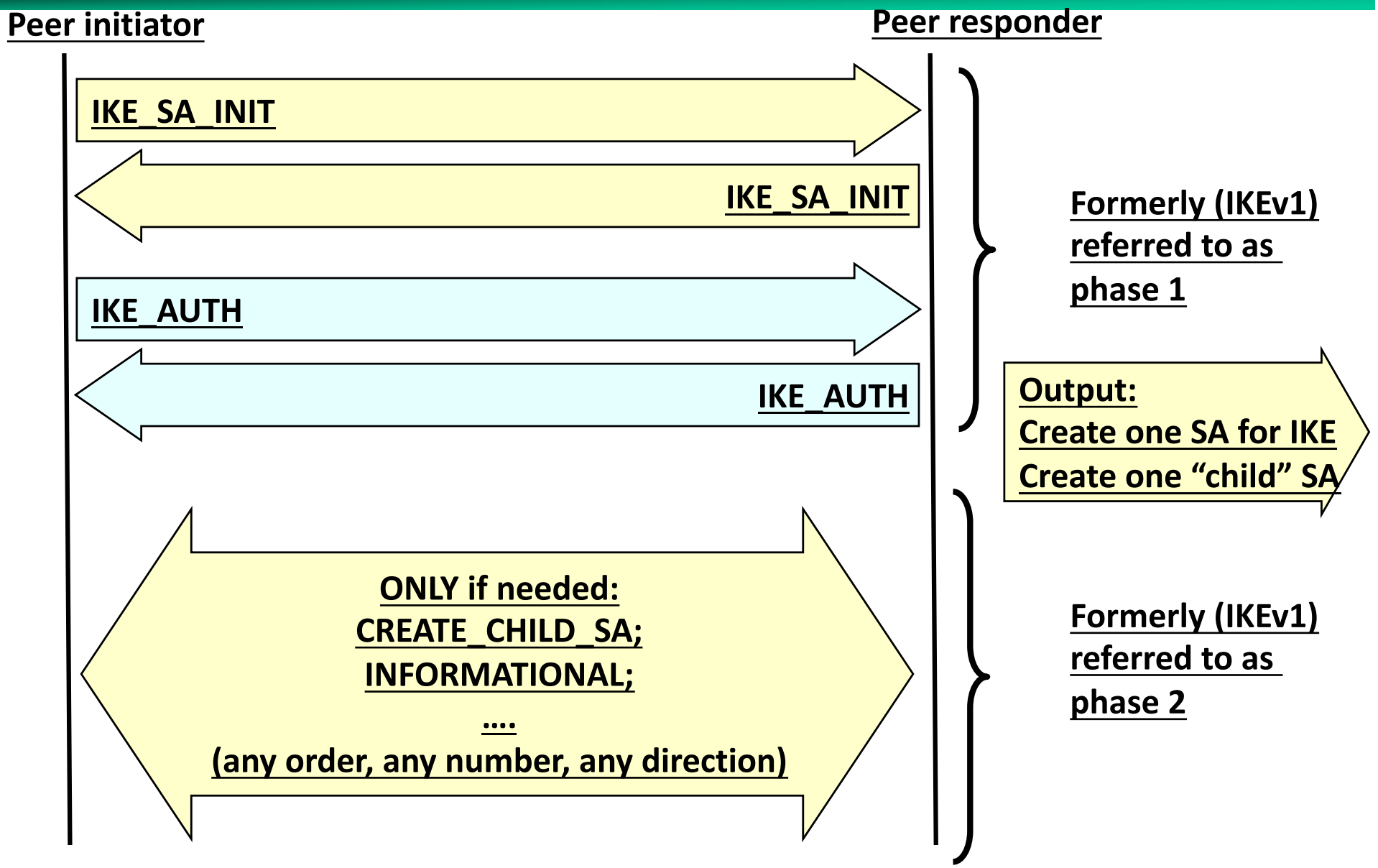
INPUT PACKET

1. **Match SPI, IP SRC and IP dst to retrieve a SA**
2. **The packet is processed**
3. **The SPD is checked to to verify associated policies**

Rationale for IKE

- shared state must be maintained between source and sink
  - » Which security services (AH, ESP)
  - » Which Crypto algorithms
  - » Which crypto keys
- Manual maintenance not scalable
  - » Partially OK only for small scale VPNs
  - » In any case, weak approach
    - » Infinite lifetime SA → no rekeying!
- IKE = Internet Key Exchange protocol
  - » Goal: dynamically establish and maintain SA
  - » IKE now (december 2005, RFC 4306) in version 2
    - » Replaces protocols specified in RFCs 2407, 2408, 2409 (IKE, ISAKMP, DOI)
    - » IKEv2 quite different (and much cleaner!!) than former specifications

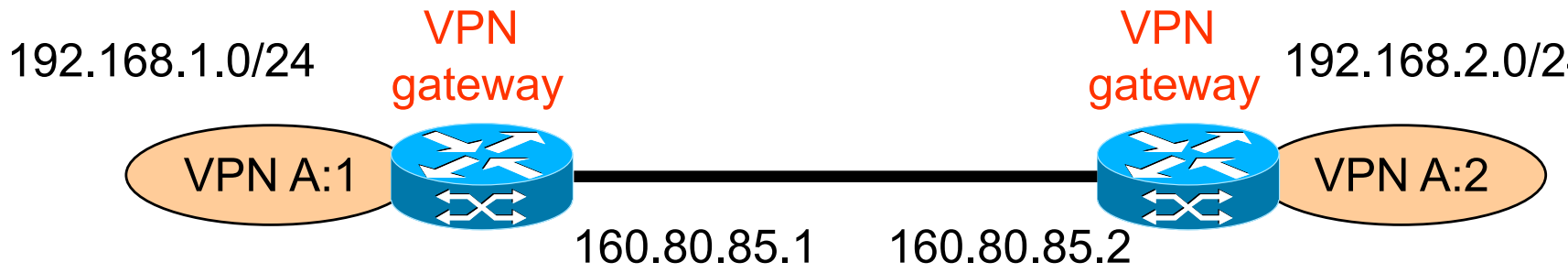
IKE phases at a glance



IKE SA and CHILD SA

- **IKE SA:**
  - » **Security association to exchange IKE messages**
- **CHILD SA**
  - » **Security association to exchange data messages**
    - » **Making use of AH or ESP**
  - » **Many CHILD SA may be set up between two peers**

VPN IPSEC Tunnel Mode (G2G)- (No IKE)



SA:

- **SPI: 0x01**
- **ESP Key: 0xf6ddb555acfd9d77b03ea3843f2653255afe8eb5573965df**
- **Src IP: 160.80.85.1**
- **Dst IP: 160.80.85.2**
- **Mode: Tunnel**

SP:

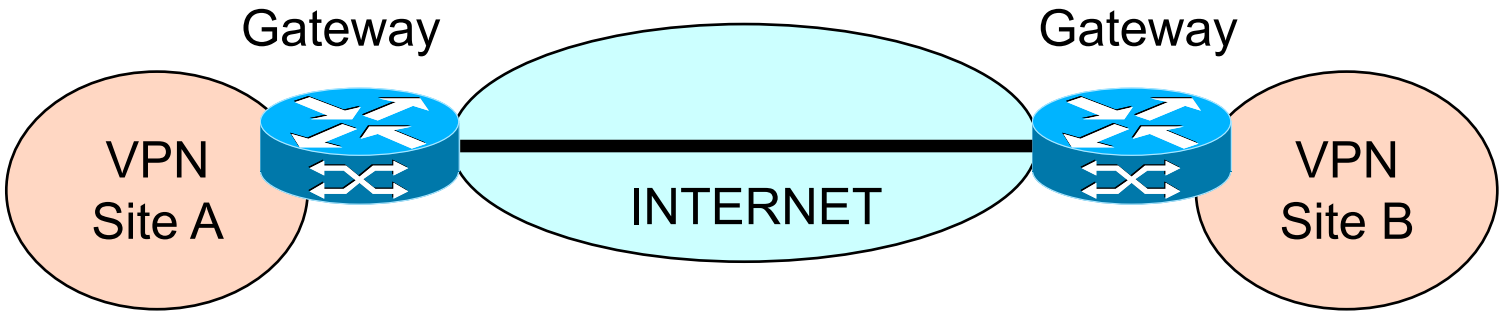
- **Net_id/mask sorgente 192.168.1.0/24**
- **Net_id/mask 192.168.2.0/24**
- **Porta sorgente / destinazione: any**
- **Direzione (in/out): out**
- **Azione da intraprendere (ipsec/discard/none): ipsec**
- **Protocollo di sicurezza (ah/esp/ipcomp): esp**
- **Modalità di incapsulamento (transport/tunnel): tunnel**
- **IP sorgente e destinazione del tunnel (solo tunnel mode): 160.80.85.1-160.80.85.2**

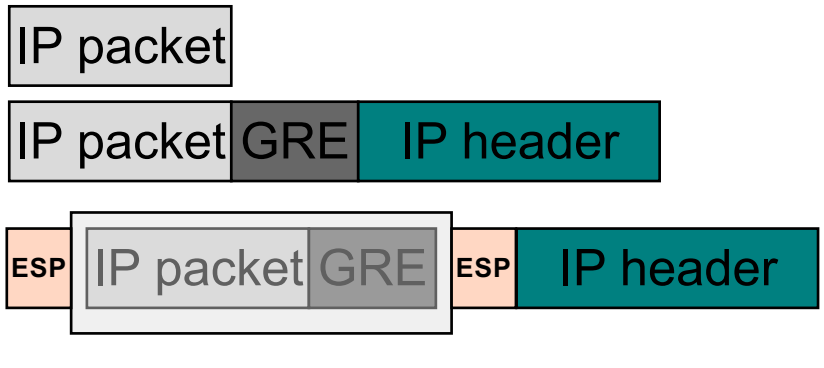
How do I transfer multicast/broadcast traffic?

IP over GRE over IPsec (transport)

IP over GRE over IPsec (transport)

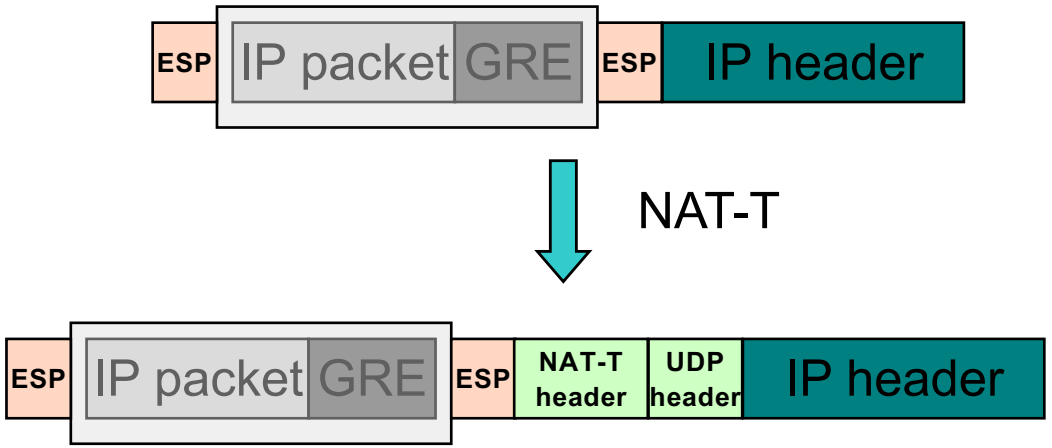
- **GRE è un protocollo che principalmente definisce solo un formato di incapsulamento; l'header GRE ha un campo protocol_type da due byte che permette anche l'incapsulamento di pacchetti IP, MPLS, etc.**
- **È utilizzato per fare dei tunnel IP**
- **Il tunnel è visibile all'utente come una scheda virtuale tun gestita dal modulo GRE del kernel**
- **Nel nostro caso,**
  - **GRE incapsula un pacchetto IP (anche multicast) da mandare sulla VPN remota**
  - **GRE genera un pacchetto IP esterno con indirizzi ip pubblici di sorgente e destinazione**
  - **Questo pacchetto IP è unicast e con indirizzi pubblici quindi trattabile da IPSec in transport mode**
- **Infine la soluzione GRE over IPSec è anche utilizzabile per supportare il dynamic routing. Pertanto offre più servizi rispetto a IPSec tunnel mode, sebbene aumenti l'overhead**





IPSec VPN vs NAT e Dynamic IP

- Stesso scenario del caso user-space VPN
- IPSec assicura l'integrità informativa di tutto il pacchetto (con AH) o della sola parte di payload (con ESP)
- Nel caso di impiego di AH, la modifica dell'header IP apportata dal NAT è vista come una alterazione dell'integrità e quindi in ricezione il pacchetto è scartato
- Con ESP questo problema non sussiste, tuttavia normalmente il NAT è un NPAT (network and port address translation) e se ESP cifra il payload, l'informazione di porta è inaccessibile ed il NPAT non inoltra il pacchetto
- Soluzione NAT-T (NAT Traversal) : il payload IP del pacchetto IPSec ESP è incapsulato in datagramma UDP con header in chiaro
- Su questo datagramma il NAT riesce ad operare
- È richesto un accordo a priori fra le parti e la disponibilità della funzione di NAT-T su entrambe le parti



GRE/IPSEC Cisco LAB with static SAs

R1 configuration

```
!Phase A: send all packets toward LAN B within the GRE tunnel
! Step 1: configure IP addresses
! IP address configuration
interface FastEthernet1/0
          ip address 160.0.0.1 255.255.255.0
          no shut

interface FastEthernet2/0
          ip address 192.168.0.1 255.255.255.0
          no shut

! Step 2: configure GRE tunnel interface
interface Tunnel0
          ip address 10.0.12.1 255.255.255.0
          tunnel source 160.0.0.1
          tunnel destination 160.0.0.2

! Step 3: configure a route via Tunnel0
ip route 192.168.1.0 255.255.255.0 Tunnel0
```

GRE/IPSEC LAB with static SAs

R1 configuration

```
!Phase B: configure IP SEC
!step 1: create an ACL for the IPSEC outbound policy
access-list 100 permit ip host 160.0.0.1 host 160.0.0.2

!step 2: create a transform set
crypto ipsec transform-set myts esp-aes

!step 3: create a crypto map (XXX for r2 swap SPI_inbound and SPI_outbound)
crypto map mycmap 1 ipsec-manual
          set peer 160.0.0.2
          set session-key inbound esp 1000 cipher 7a8ec0d7f95b01d46758830ba0de280f
          set session-key outbound esp 1001 cipher 7a8ec0d7f95b01d46758830ba0de280f
          set transform-set myts
          match address 100

!step 4: attach my new crypto map to F1/0 (the out interface)
interface FastEthernet1/0
          crypto map mycmap
```

IPSEC configuration with IKE

- IKE automatically negotiates IPsec security associations (SAs) and enables IPsec secure communications without costly manual preconfiguration
- Benefits
  - » Eliminates the need to manually specify all the IPsec security parameters in the crypto maps at both peers
  - » Allows you to specify a lifetime for the IPsec SA
  - » Allows encryption keys to change during IPsec sessions
  - » Allows IPsec to provide antireplay services
  - » Permits certification authority (CA) support for a manageable, scalable IPsec implementation
  - » Allows dynamic authentication of peers

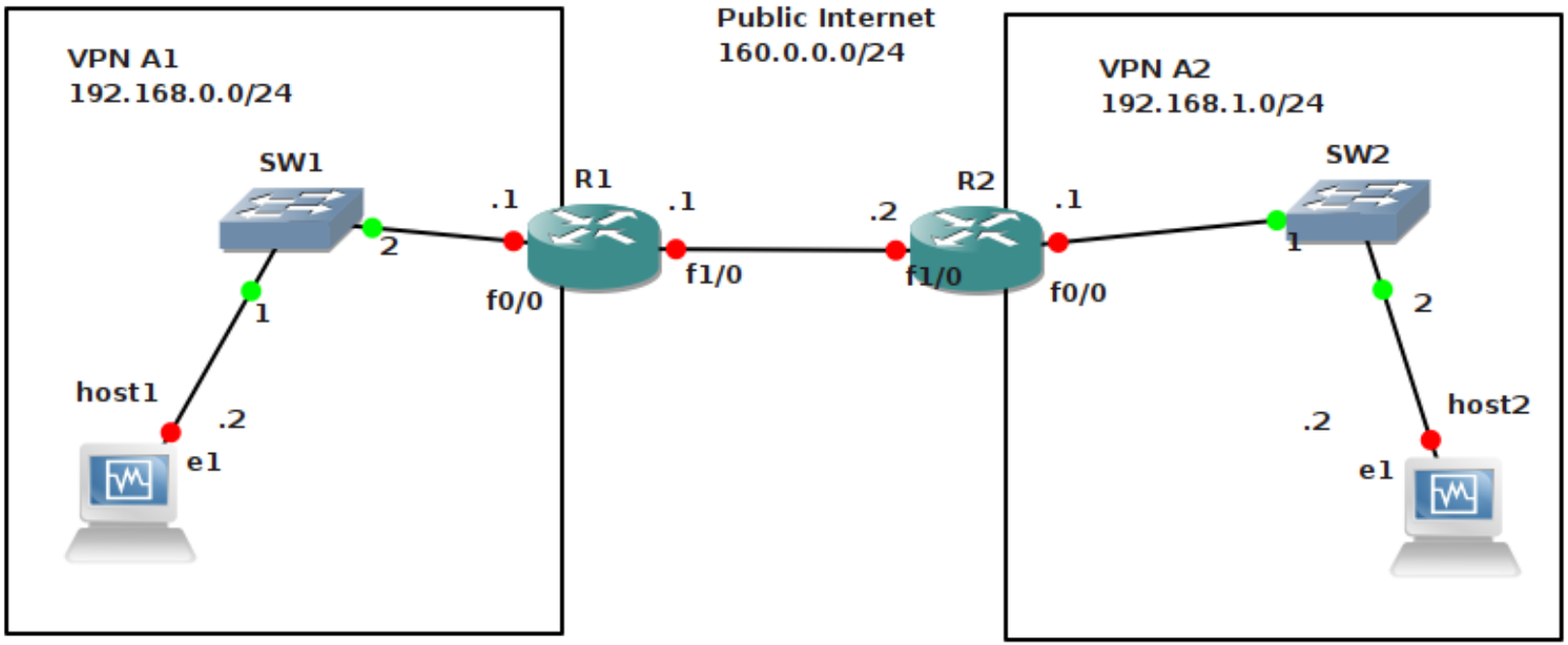
IPSEC configuration with IKE

- Each IKE negotiation begins by agreement of both peers on a common (shared) IKE policy.
  - » This policy states which security parameters will be used to protect subsequent IKE negotiations and mandates how the peers are authenticated.
- After the two peers agree upon a policy, the security parameters of the policy are identified by an SA established at each peer, and these SAs apply to all subsequent IKE traffic during the negotiation
- When the IKE negotiation begins, IKE searches for an IKE policy that is the same on both peers
  - » The peer that initiates the negotiation will send all its policies to the remote peer, and the remote peer will try to find a match.
  - » The remote peer looks for a match by comparing its own highest priority policy against the policies received from the other peer.
  - » The remote peer checks each of its policies in order of its priority (highest priority first) until a match is found.
  - » A match is made when both policies from the two peers contain the same encryption, hash, authentication, and Diffie-Hellman parameter values.

IPSEC IKE

- If a match is found, IKE will complete negotiation, and IPsec security associations will be created.
- If no acceptable match is found, IKE refuses negotiation and IPsec will not be established.
- IKE authentication type can be one of the following:
  - » RSA Signatures
  - » RSA Encrypted Nonces
  - » Preshared Keys

IKE LAB: pre-shared secret



Same LAB, same IP/GRE configuration!

IPSEC/IKE LAB: pre-shared keys

R1 configuration

```
!configure IKE policy
crypto isakmp policy 10
         encryption aes 256
         hash sha
         authentication pre-share
         group 5
         lifetime 180

!configure IKE identity identità ike
crypto isakmp key "blablabla1234" address 160.0.0.2

!configure a transform set
crypto ipsec transform-set myts2 esp-3des esp-md5-hmac

!configure a crypto map and reference the ike policy
crypto map mycmap2 10 ipsec-isakmp
         set peer 160.0.0.2
         set transform-set myts2
         match address 100

!attach crypto map to out interface
fastEthernet 1/0
         crypto map mycmap2
```